AD-A182 065

AFWAL-TR-86-4006
Volume VII
Part 3

INTEGRATED INFORMATION
SUPPORT SYSTEM (IISS)
Volume VII - Communications Subsystem
Part 3 - VAX IPC Product Specification

General Electric Company
Production Resources Consulting
One River Road
Schenectady, New York 12345

DTIC
ELECTE
JUL 0 6 1987
D

Final Report for Period 22 September 1980 - 31 July 1985

November 1985

Approved for public release; distribution is unlimited.

MATERIALS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AFB, OH 45433-6533

NOTICE

When Government drawings specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated. furnished. or in any way supplied the said drawings. specifications. or other data. is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use. or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS) At NTIS. it will be available to the general public. including foreign nations.

This technical report has been reviewed and is approved for publication.

DAVID L. JUDSON, PROJECT MANAGER
AFWAL/MLTC
WRIGHT PATTERSON AFB OH 45433

DATE 5 Aug 1986

FOR THE COMMANDER:

GERALD C. SHUMAKER, BRANCH CHIEF
AFWAL/MLTC
WRIGHT PATTERSON AFB OH 45433

DATE 7 Aug 86

A182 685

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
Unclassified	
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT
	Approved for public release:
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	distribution is unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
	AFWAL-TR-86-4006 Vol VII, Part 3

6a. NAME OF PERFORMING ORGANIZATION	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
General Electric Company Production Resources Consulting		AFWAL/MLTC
6c. ADDRESS (City, State and ZIP Code)		7b. ADDRESS (City, State and ZIP Code)
1 River Road Schenectady, NY 12345		WPAFB, OH 45433-6533

8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
Materials Laboratory Air Force Systems Command, USAF	AFWAL/MLTC	F33615-80-C-5155

8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS			
Wright-Patterson AFB, Ohio 45433	PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
	78011F	7500	62	01

11. TITLE (Include Security Classification)
(See Reverse)

12. PERSONAL AUTHOR(S)
Phillips, Donna

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Final Technical Report	22 Sept 1980 - 31 July 1985	1985 November	121

16. SUPPLEMENTARY NOTATION

ICAM Project Priority 6201

The computer software contained herein are theoretical and/or references that in no way reflect Air Force-owned or -developed computer software.

17.	COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB GR	
1308	0905		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This document defines the Product Specification for the Interprocess Communication Primitives (IPC) configuration item of the Integrated Information Support System (IISS) Test Bed. The Test Bed is an integrated hardware/software environment in which various Integrated Computer Aided Manufacturing (ICAM) Program applications will function cooperatively.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
David L. Judson	513-255-6076	AFWAL/MLTC

DD FORM 1473, 83 APR   EDITION OF 1 JAN 73 IS OBSOLETE.

11. Title

Integrated Information Support System (IISS)
Vol VII - Communications Subsystem
Part 3 - VAX IPC Product Specification

A S D 86 1438
17 Jul 1986

PREFACE

This product specification covers the work performed under Air Force Contract F33615-80-C-5155 (ICAM Project 6201). This contract is sponsored by the Materials Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Gerald C. Shumaker, ICAM Program Manager, Manufacturing Technology Division, through Project Manager, Mr. David Judson. The Prime Contractor was Production Resources Consulting of the General Electric Company, Schenectady, New York, under the direction of Mr. Alan Rubenstein. The General Electric Project Manager was Mr. Myron Hurlbut of Industrial Automation Systems Department, Albany, New York.

Certain work aimed at improving Test Bed Technology has been performed by other contracts with Project 6201 performing integrating functions. This work consisted of enhancements to Test Bed software and establishment and operation of Test Bed hardware and communications for developers and other users. Documentation relating to the Test Bed from all of these contractors and projects have been integrated under Project 6201 for publication and treatment as an integrated set of documents. The particular contributors to each document are noted on the Report Documentation Page (DD1473). A listing and description of the entire project documentation system and how they are related is contained in document FTR620100001, Project Overview.

The subcontractors and their contributing activities were as follows:

TASK 4.2

Subcontractors	Role
Boeing Military Aircraft Company (BMAC)	Reviewer.
D. Appleton Company (DACOM)	Responsible for IDEF support, state-of-the-art literature search.
General Dynamics/ Ft. Worth	Responsible for factory view function and information models.

iii

Subcontractors	Role
Illinois Institute of Technology	Responsible for factory view function research (IITRI) and information models of small and medium-size business.
North American Rockwell	Reviewer.
Northrop Corporation	Responsible for factory view function and information models.
Pritsker and Associates	Responsible for IDEF2 support.
SofTech	Responsible for IDEF0 support.

TASKS 4.3 - 4.9 (TEST BED)

Subcontractors	Role
Boeing Military Aircraft Company (BMAC)	Responsible for consultation on applications of the technology and on IBM computer technology.
Computer Technology Associates (CTA)	Assisted in the areas of communications systems, system design and integration methodology, and design of the Network Transaction Manager.
Control Data Corporation (CDC)	Responsible for the Common Data Model (CDM) implementation and part of the CDM design (shared with DACOM).
D. Appleton Company (DACOM)	Responsible for the overall CDM Subsystem design integration and test plan, as well as part of the design of the CDM (shared with CDC). DACOM also developed the Integration Methodology and did the schema mappings for the Application Subsystems.

iv

Subcontractors	Role
Digital Equipment Corporation (DEC)	Consulting and support of the performance testing and on DEC software and computer systems operation.
McDonnell Douglas Automation Company (McAuto)	Responsible for the support and enhancements to the Network Transaction Manager Subsystem during 1984/1985 period.
On-Line Software International (OSI)	Responsible for programming the Communications Subsystem on the IBM and for consulting on the IBM.
Rath and Strong Systems Products (RSSP) (In 1985 became McCormack & Dodge)	Responsible for assistance in the implementation and use of the MRP II package (PIOS) that they supplied.
SofTech, Inc.	Responsible for the design and implementation of the Network Transaction Manager (NTM) in 1981/1984 period.
Software Performance Engineering (SPE)	Responsible for directing the work on performance evaluation and analysis.
Structural Dynamics Research Corporation (SDRC)	Responsible for the User Interface and Virtual Terminal Interface Subsystems.

Other prime contractors under other projects who have contributed to Test Bed Technology, their contributing activities and responsible projects are as follows:

Contractors	ICAM Project	Contributing Activities
Boeing Military Aircraft Company (BMAC)	1701, 2201, 2202	Enhancements for IBM node use. Technology Transfer to Integrated Sheet Metal Center (ISMC).

Contractors	ICAM Project	Contributing Activities
Control Data Corporation (CDC)	1502, 1701	IISS enhancements to Common Data Model Processor (CDMP).
D. Appleton Company (DACOM)	1502	IISS enhancements to Integration Methodology.
General Electric	1502	Operation of the Test Bed and communications equipment.
Hughes Aircraft Company (HAC)	1701	Test Bed enhancements.
Structural Dynamics Research Corporation (SDRC)	1502, 1701, 1703	IISS enhancements to User Interface/Virtual Terminal Interface (UI/VTI).
Systran	1502	Test Bed enhancements. Operation of Test Bed.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

SECTION 1

SCOPE

1.1 Identification

This specification establishes the 'as built' design of the Interprocess Communication Primitives (IPC's).

1.2 Functional Summary

The IPC's are the lowest level of common services in the IISS. Their function is to supply a standard interface to operating system services normally required by complex application systems. The implementation of these routines, therefore, is highly system dependent.

The services they support are passing messages between two programs, starting and stoping a timer, and responding to a series of asynchronous events.

SECTION 2

DOCUMENTS

2.1 Reference Documents

The following pertinent reference materials are available at the ICAM Program Office.

1. Interim Reports

2. Life Cycle Documents
    (a) ITR620150002U Project Scope
    (b) PMP620150000 Master Plan and Schedule
    (c) SAD620150000 State-of-the-art Review
    (d) SRD620140000 System Requirements Document
    (e) SDS620140000 System Design Specifications
    (f) DS6201430000 Development Specification - Communications Subsystem

The following reference materials are available from Digital Equipment Corporation.
    (a) VAX/VMS I/O User's Guide (Volume 1), Order No. AA-M540B-TE
    (b) VAX/VMS I/O User's Guide (Volume 2), Order No. AA-M541B-TE
    (c) VAX COBOL Language Reference Manual, Order No. AA-H631C-TE
    (d) VAX-11 FORTRAN Language Reference Manual, Order No. AA-D034C-TE

2.2 Terms and Abbreviations

All the arguments for the IPC's are described in DS 620140004 for the Communication Subsystem; however, the more important ones are reiterated here.

1. Input Mailbox - the logical name for a section of memory from which a primitive obtains a message that was send to a program

2. Target Mailbox - the logical name for a section of memory into which a primitive stores a message

3. Event Block - a block of contiguous memory, local to a program, into which primitives store system dependent

information needed to perform their tasks

SECTION 3

REQUIREMENTS

3.1 Structural Descriptions

The description of each IPC along with its calling sequence, its inputs and its outputs is given is Sections 3.2.7 and 3.5 of the Communication Subsystem Development Specification, DS620140004. Thus, only a list of the IPC's and their functions will be presented here.

3.1.1 Create a Mailbox

The routine CRTMBX creates a mailbox (global storage in memory) through which a program may receive messages from another program executing on the same computer at the same time.

3.1.2 Send a Message to Another Program

The routine SNDMSG sends a message to another program executing on the same computer at the same time by placing it in the input mailbox (memory) allocated to the other program.

3.1.3 Receive a Message from Another Program

The routine RCVMSG records the fact that the program will accept messages sent from other programs to its input mailbox. The routine returns to the program allowing it to execute while other programs send messages.

3.1.4 Get a Message from Another Program

The routine GETMSG removes a message from the input mailbox (memory) of a program and moves it to the given buffer.

3.1.5 Delete a Mailbox

The routine DELMBX removes the capability of receiving messages through the given input mailbox. Whether the memory is released for other uses at this time depends upon the computer and operating system under which IISS runs.

3.1.6 Release an Event Block

The routine RELEVB reinitializes an event block (local

program storage) enabling it to be used with other primitives. It is needed to clear an event block when the same event block is used with different target mailboxes.

3.1.7 Start a Timer

The routine SETTIM invokes a timing mechanism that will cause an event when the time interval has elapsed. After noting the time, the routine returns to the program allowing it to continue executing.

3.1.8 Stop a Timer

The routine CNLTIM terminates the timing mechanism initiated by the SETTIM routine.

3.1.9 Wait for an Event to Occur

The routine WAITnn (where nn is the maximum number of events possible) waits for the completion of one of the outstanding requests that are associated with the list of event blocks. The program is suspended until an event occurs.

3.1.10 Terminate a Run

The routine ENDRUN terminates the executing of the program.

3.2 Functional Flow Description

Since the implementation of the IPC's are system dependent, the functional flow description must be described in context of the computer and operating system. The IPC's were implemented on the IBM 3084 under CICS and under MVS. They were implemented on the Honeywell Level 6 under Mod 400 and on the VAX under VMS.

3.3 Interfaces

On the IBM, under CICS, the IPC's were implemented using some CICS command level and some macro level system interfaces. The command level was written in COBOL, while the macro level was in assembler. This mixed mode format was required because not all the necessary functions were available at the command level. There is no mailbox facility under CICS, so the concept of mailbox had to designed and implemented from scratch.

Under MVS, the IPC's were implemented in assembler as one task with different entry points for each routine. Again, there is no mailbox facility under MVS. The design developed for CICS was reimplemented for MVS.

On the Honeywell Level 6, the IPC's were again a mixture of COBOL and assembler because the only interface into the system services is through assembler. The Mod 400 operating system supports a mailbox facility so calls to it were used in the implementation of the IPC's.

The VAX also has a mailbox facility and its system services are accessible through high level languages. Therefore, the IPC's on the VAX were implemented using a conbination of COBOL and FORTRAN.

3.4 Interrupts

This section does not apply to the IPS's.

3.5 Timing and Sequence Description

Timing and sequencing is only relevant to the Wait for an Event to Occur primitive, and then only when it is waiting on more than one event. If more than one event has occurred, the primitive will report to the program the one that has the highest priority based on event number.

3.6 Special Control Features

The Interprocess Communication Primitives do not include any special control features as defined in the ICAM Documentations Standards.

3.7 Storage Allocation

3.7.1 Data Base Definition

3.7.1.1 File Description

The IPC's do not use any files.

3.7.1.2 Table Description

The IPC's do not use any tables.

3.7.1.3 Item and Constant Description

The event block contains two variables that are common across all implementations. The first is the event type which indicates whether the block is being used by an IPC receive message, a timer, or a receive for the Communications Subsystem. The second variable indicates the possible states of the event--no event outstanding, event outstanding but not completed, event completed but not requested by the program.

3.7.2 CPC Relationship

The event block is used with every IPC except ENDRUN and LOCKEF.

3.8 Object Code Creation

To create the IPC's for the CICS environment requires the CICS preprocessor, the COBOL compiler, the Assembler and the standard link editor to create a load module that can be referenced when linking main modules. Except for the CICS preprocessor, the same mechanism applies for creating the IPC load library under MVS.

To create the IPC library on the Honeywell Level 6 requires the COBOL compiler, the Assembler and the standard linker.

To create the IPC library on the VAX requires the COBOL and FORTRAN compilers and the standard linker.

3.9 Adaption Data

The IPC's are system dependent; therefore, they must be reimplemented for each computer and/or operating system.

3.10 Detail Design Description

3.10.1 Main Program List

The following is a list of all "Main Programs" which are modules that are not called by any other module being documented here. These modules are either program entry points or, if they are hooked into another set of programs via subroutine calls, they are the points the external programs can call and therefore enter through. To differentiate between the two types of entry points, look at the individual Module

Documentation (section 3.10.8) and look at Module Type for each
of the Main Program modules listed. Note whether the routine
is a Program, Subroutine, or Function. If it is a Program, it
is truly a main program entry point. If not, then it is merely
called by other programs not being documented here.

IPC Main Program List

Module Name	Purpose
CNLTIM	CANCEL A TIMER
CRTMBX	CREATE A MAILBOX
DELMBX	DELETE A MAILBOX
ENDRUN	STOP THE PROGRAM
GETMSG	GET A MESSAGE FROM ANOTHER PROGRAM
LOCKEF	LOCK EVENT FLAG
RCVMSG	RECEIVE A MESSAGE FROM ANOTHER PROGRAM
RELEVB	RELEASE A TARGET EVENT BLOCK (CLEAR IT) AND
SETTIM	START A TIMER
SNDMSG	SEND A MESSAGE TO ANOTHER PROGRAM
WAITO1	WAIT FOR AN EVENT TO OCCUR
WAITO2	WAIT FOR AN EVENT TO OCCUR
WAITO3	WAIT FOR AN EVENT TO OCCUR

3.10.2 Module List

The following is a list of all the modules being documented here along with their purpose. Each module has a unique name, no matter what language it was written in.

IPC Module List

Module Name	Purpose
CNLTIM	CANCEL A TIMER
CNLTMR	CANCEL THE TIMER
CRTMBX	CREATE A MAILBOX
DELMBX	DELETE A MAILBOX
ENDRUN	STOP THE PROGRAM
ERRFTL	PROCESS FATAL ERROR
ERRPRO	PROCESS ERROR
FREVTF	FREE THE EVENT FLAG
GETMSG	GET A MESSAGE FROM ANOTHER PROGRAM
GETPNM	GET PROCESS NAME
LOCKEF	LOCK EVENT FLAG
RCVMSG	RECEIVE A MESSAGE FROM ANOTHER PROGRAM
RDMALF	READ THE MAILBOX FOR EVENT FLAG SET
RELEVB	RELEASE A TARGET EVENT BLOCK (CLEAR IT) AND
SASGCH	ASSIGN A CHANNEL TO A MAILBOX
SCRTMB	CREATE A MAILBOX
SDEACH	DEASSIGN CHANNEL FROM THE MAILBOX
SDELMB	DELETE THE MAILBOX
SETTIM	START A TIMER
SETTMR	SET A TIMER

IPC Module List

Module Name	Purpose
SNDMSG	SEND A MESSAGE TO ANOTHER PROGRAM
SWEVTF	WAIT FOR ONE OF POSSIBLY MANY EVENTS
VALMBE	VALIDATE THAT MAILBOX ALREADY EXISTS
WAIT01	WAIT FOR AN EVENT TO OCCUR
WAIT02	WAIT FOR AN EVENT TO OCCUR
WAIT03	WAIT FOR AN EVENT TO OCCUR
WRTERR	WRITE MESSAGE TO MAILBOX
WRTMAI	WRITE THE MESSAGE INTO THE MAILBOX

3.10.3 External Routines List

The following is a list of all routines or functions not documented here that are called by modules that are documented here. The first caller, in alphabetical order, is listed as well. The specification in which any module is documented may be found in the Module Documentation Index (Document Number CM 620100001). See section 3.10.6 for a list of the modules that call each of these external routines.

IPC External Routines List

Module Name	First User
LIB$FREE_EF	SWEVTF
LIB$GET_EF	WRTERR
SYS$ASSIGN	SASGCH
SYS$BINTIM	WRTERR
SYS$CANTIM	CNLTMR
SYS$CLREF	SWEVTF
SYS$CREMBX	SCRTMB
SYS$DASSGN	WRTERR
SYS$DELMBX	SDELMB
SYS$GETDEV	VALMBE
SYS$GETJPI	GETPNM
SYS$QIO	RDMALF
SYS$QIOW	WRTERR
SYS$SETIMR	SETTMR
SYS$SETRWM	SCRTMB
SYS$WAITFR	WRTERR
SYS$WFLOR	SWEVTF

3.10.4   Include File List

The following is a list of all include files called in
by modules being documented here. Each include file has a
unique name regardless of the language being used. The purpose
of each include file is listed as well. A more complete
description of each include file is given in section 3.10.9.
The purpose listed is the one that is in the source code of the
include file.

A purpose of "**** PURPOSE NOT FOUND BY STRIPPER ****"
indicates that a purpose statement was not written into the
include file itself. The most common reason for this is that
the include file comes from system libraries that were not
developed by the project, such as 'C' libraries that are
provided with the 'C' compiler.

See section 3.10.6 for a set of lists which show all
the modules which call in each of these include files.

IPC Include File List

File Name	Purpose
($JPIDEF)	**** PURPOSE NOT FOUND BY STRIPPER ****
CHKSTS	CHKSTS.INC -- CHECK STATUS
ERRPRO	PROCESS ERROR INCLUDE FILE
ERRSTS	ERRSTS.INC -- IISS ERROR CODES
ERRSTS.INF	**** PURPOSE NOT FOUND BY STRIPPER ****
KIPC	KIPC.INC -- CONSTAND BLOCK FOR IPC PRIMITIVES
MBEB01	MBEB01.INC -- MAILBOX EVENT BLOCK DESCRIPTION
MBEB02	MBEB02.INC -- MAILBOX EVENT BLOCK DESCRIPTION
MBEB03	MBEB03.INC -- MAILBOX EVENT BLOCK DESCRIPTION
MBXEBK	MBXEBK.INC -- MAILBOX EVENT BLOCK DESCRIPTION
TIMREB	TIMREB.INC -- TIME EVENT BLOCK DESCRIPTION

3.10.5 Where Include File Used List

The following lists each include file from 3.10.4 and all the modules documented in this specification which include them. The purpose of each module is listed as well.

IPC Where-include-file-used List

Include File	Module Name	Module Purpose
(\$JPIDEF)		
	GETPNM	GET PROCESS NAME
CHKSTS		
	CNLTIM	CANCEL A TIMER
	CRTMBX	CREATE A MAILBOX
	DELMBX	DELETE A MAILBOX
	ERRFTL	PROCESS FATAL ERROR
	ERRPRO	PROCESS ERROR
	GETMSG	GET A MESSAGE FROM ANOTHER PROGRAM
	RCVMSG	RECEIVE A MESSAGE FROM ANOTHER PROGRAM
	RELEVB	RELEASE A TARGET EVENT BLOCK (CLEAR IT) AND
	SETTIM	START A TIMER
	SNDMSG	SEND A MESSAGE TO ANOTHER PROGRAM
	WAIT01	WAIT FOR AN EVENT TO OCCUR
	WAIT02	WAIT FOR AN EVENT TO OCCUR
	WAIT03	WAIT FOR AN EVENT TO OCCUR
ERRPRO		
	CNLTIM	CANCEL A TIMER
	CRTMBX	CREATE A MAILBOX
	DELMBX	DELETE A MAILBOX
	GETMSG	GET A MESSAGE FROM ANOTHER PROGRAM
	RCVMSG	RECEIVE A MESSAGE FROM ANOTHER PROGRAM
	RELEVB	RELEASE A TARGET EVENT BLOCK (CLEAR IT) AND
	SETTIM	START A TIMER
	SNDMSG	SEND A MESSAGE TO ANOTHER PROGRAM
	WAIT01	WAIT FOR AN EVENT TO OCCUR
	WAIT02	WAIT FOR AN EVENT TO OCCUR
	WAIT03	WAIT FOR AN EVENT TO OCCUR

IPC Where-include-file-used List

Include File	Module Name	Module Purpose
-------	------	-------

ERRSTS

	CNLTIM	CANCEL A TIMER
	CRTMBX	CREATE A MAILBOX
	DELMBX	DELETE A MAILBOX
	GETMSG	GET A MESSAGE FROM ANOTHER PROGRAM
	RCVMSG	RECEIVE A MESSAGE FROM ANOTHER PROGRAM
	RELEVB	RELEASE A TARGET EVENT BLOCK (CLEAR IT) AND
	SETTIM	START A TIMER
	SNDMSG	SEND A MESSAGE TO ANOTHER PROGRAM
	WAITO1	WAIT FOR AN EVENT TO OCCUR
	WAITO2	WAIT FOR AN EVENT TO OCCUR
	WAITO3	WAIT FOR AN EVENT TO OCCUR

ERRSTS.INF

	CNLTMR	CANCEL THE TIMER
	GETPNM	GET PROCESS NAME
	LOCKEF	LOCK EVENT FLAG
	RDMALF	READ THE MAILBOX FOR EVENT FLAG SET
	SASGCH	ASSIGN A CHANNEL TO A MAILBOX
	SCRTMB	CREATE A MAILBOX
	SDEACH	DEASSIGN CHANNEL FROM THE MAILBOX
	SDELMB	DELETE THE MAILBOX
	SETTMR	SET A TIMER
	VALMBE	VALIDATE THAT MAILBOX ALREADY EXISTS
	WRTERR	WRITE MESSAGE TO MAILBOX
	WRTMAI	WRITE THE MESSAGE INTO THE MAILBOX

IPC Where-include-file-used List

Include File	Module Name	Module Purpose
KIPC		
	CRTMBX	CREATE A MAILBOX
	GETMSG	GET A MESSAGE FROM ANOTHER PROGRAM
	RCVMSG	RECEIVE A MESSAGE FROM ANOTHER PROGRAM
	SNDMSG	SEND A MESSAGE TO ANOTHER PROGRAM
MBEBO1		
	WAITO1	WAIT FOR AN EVENT TO OCCUR
	WAITO2	WAIT FOR AN EVENT TO OCCUR
	WAITO3	WAIT FOR AN EVENT TO OCCUR
MBEBO2		
	WAITO2	WAIT FOR AN EVENT TO OCCUR
	WAITO3	WAIT FOR AN EVENT TO OCCUR
MBEBO3		
	WAITO3	WAIT FOR AN EVENT TO OCCUR
MBXEBK		
	CRTMBX	CREATE A MAILBOX
	DELMBX	DELETE A MAILBOX
	GETMSG	GET A MESSAGE FROM ANOTHER PROGRAM
	RCVMSG	RECEIVE A MESSAGE FROM ANOTHER PROGRAM
	RELEVB	RELEASE A TARGET EVENT BLOCK (CLEAR IT) AND
	SNDMSG	SEND A MESSAGE TO ANOTHER PROGRAM

IPC Where-include-file-used List

Include File	Module Name	Module Purpose
TIMREB		
	CNLTIM	CANCEL A TIMER
	SETTIM	START A TIMER

3.10.6 Where External Routine Used List

The following lists each external function or routine
listed in 3.10.3 and all the documented modules which call it.
The purpose of each module is listed as well.

IPC Where-external-routine-used List

System Module	Module Name	Module Purpose

LIB$FREE_EF

	FREVTF	FREE THE EVENT FLAG
	SWEVTF	WAIT FOR ONE OF POSSIBLY MANY EVENTS
	WRTERR	WRITE MESSAGE TO MAILBOX

LIB$GET_EF

	LOCKEF	LOCK EVENT FLAG
	RDMALF	READ THE MAILBOX FOR EVENT FLAG SET
	SETTMR	SET A TIMER
	WRTERR	WRITE MESSAGE TO MAILBOX

SYS$ASSIGN

| | SASGCH | ASSIGN A CHANNEL TO A MAILBOX |
| | WRTERR | WRITE MESSAGE TO MAILBOX |

SYS$BINTIM

| | SETTMR | SET A TIMER |
| | WRTERR | WRITE MESSAGE TO MAILBOX |

SYS$CANTIM

| | CNLTMR | CANCEL THE TIMER |

SYS$CLREF

| | SWEVTF | WAIT FOR ONE OF POSSIBLY MANY EVENTS |

IPC Where-external-routine-used List

System Module	Module Name	Module Purpose

SYS$CREMBX

      SCRTMB   CREATE A MAILBOX

SYS$DASSGN

      SDEACH   DEASSIGN CHANNEL FROM THE MAILBOX

      WRTERR   WRITE MESSAGE TO MAILBOX

SYS$DELMBX

      SDELMB   DELETE THE MAILBOX

SYS$GETDEV

      VALMBE   VALIDATE THAT MAILBOX ALREADY EXISTS

SYS$GETJPI

      GETPNM   GET PROCESS NAME

SYS$QIO

      RDMALF   READ THE MAILBOX FOR EVENT FLAG SET

SYS$QIOW

      WRTERR   WRITE MESSAGE TO MAILBOX

      WRTMAI   WRITE THE MESSAGE INTO THE MAILBOX

IPC Where-external-routine-used List

System Module	Module Name	Module Purpose

SYS$SETIMR
| | SETTMR | SET A TIMER |
| | WRTERR | WRITE MESSAGE TO MAILBOX |

SYS$SETRWM
| | SCRTMB | CREATE A MAILBOX |

SYS$WAITFR
| | WRTERR | WRITE MESSAGE TO MAILBOX |

SYS$WFLOR
| | SWEVTF | WAIT FOR ONE OF POSSIBLY MANY EVENTS |

3.10.7 Main Program Parts List

The following lists each Main Program listed in 3.10.1 and all the modules which are called either by that module itself or by any of the documented modules which it calls. It is possible for a non-main module to be listed more that once if it is called by multiple modules. The called modules, in this case known as program parts, are marked as to whether they are documented here. If so, the phrase "well-defined module" appears by the module name, if not it is an "external "routine". The Purpose of the Main Program module is listed as well.

IPC Main Program Parts List

Main Pgm Name	Module Name	Module Type
CNLTIM		Purpose--›CANCEL A TIMER
	CNLTMR	Well-defined module
	ERRFTL	Well-defined module
	ERRPRO	Well-defined module
	FREVTF	Well-defined module
	GETPNM	Well-defined module
	LIB$FREE_EF	External routine
	LIB$GET_EF	External routine
	SYS$ASSIGN	External routine
	SYS$BINTIM	External routine
	SYS$CANTIM	External routine
	SYS$DASSGN	External routine
	SYS$GETJPI	External routine
	SYS$QIOW	External routine
	SYS$SETIMR	External routine
	SYS$WAITFR	External routine
	WRTERR	Well-defined module

IPC Main Program Parts List

Main Pgm Name	Module Name	Module Type
CRTMBX	Purpose--›CREATE A MAILBOX	
	ERRFTL	Well-defined module
	ERRPRO	Well-defined module
	GETPNM	Well-defined module
	LIB$FREE_EF	External routine
	LIB$GET_EF	External routine
	SCRTMB	Well-defined module
	SYS$ASSIGN	External routine
	SYS$BINTIM	External routine
	SYS$CREMBX	External routine
	SYS$DASSGN	External routine
	SYS$GETDEV	External routine
	SYS$GETJPI	External routine
	SYS$QIOW	External routine
	SYS$SETIMR	External routine
	SYS$SETRWM	External routine
	SYS$WAITFR	External routine
	VALMBE	Well-defined module
	WRTERR	Well-defined module

IPC Main Program Parts List

Main Pgm Name	Module Name	Module Type
--------	------	------
DELMBX	Purpose--›DELETE A MAILBOX	
	ERRFTL	Well-defined module
	ERRPRO	Well-defined module
	GETPNM	Well-defined module
	LIB$FREE_EF	External routine
	LIB$GET_EF	External routine
	SDEACH	Well-defined module
	SDELMB	Well-defined module
	SYS$ASSIGN	External routine
	SYS$BINTIM	External routine
	SYS$DASSGN	External routine
	SYS$DELMBX	External routine
	SYS$GETJPI	External routine
	SYS$QIOW	External routine
	SYS$SETIMR	External routine
	SYS$WAITFR	External routine
	WRTERR	Well-defined module

IPC Main Program Parts List

Main Pgm Name	Module Name	Module Type
GETMSG		Purpose--›GET A MESSAGE FROM ANOTHER PROGRAM
	ERRFTL	Well-defined module
	ERRPRO	Well-defined module
	FREVTF	Well-defined module
	GETPNM	Well-defined module
	LIB$FREE_EF	External routine
	LIB$GET_EF	External routine
	SYS$ASSIGN	External routine
	SYS$BINTIM	External routine
	SYS$DASSGN	External routine
	SYS$GETJPI	External routine
	SYS$QIOW	External routine
	SYS$SETIMR	External routine
	SYS$WAITFR	External routine
	WRTERR	Well-defined module

IPC Main Program Parts List

Main Pgm Name	Module Name	Module Type
LOCKEF		Purpose--›LOCK EVENT FLAG
	ERRFTL	Well-defined module
	ERRPRO	Well-defined module
	GETPNM	Well-defined module
	LIB$FREE_EF	External routine
	LIB$GET_EF	External routine
	SYS$ASSIGN	External routine
	SYS$BINTIM	External routine
	SYS$DASSGN	External routine
	SYS$GETJPI	External routine
	SYS$QIOW	External routine
	SYS$SETIMR	External routine
	SYS$WAITFR	External routine
	WRTERR	Well-defined module

IPC Main Program Parts List

Main Pgm Name	Module Name	Module Type
RCVMSG		Purpose--›RECEIVE A MESSAGE FROM ANOTHER PROGRAM
	ERRFTL	Well-defined module
	ERRPRO	Well-defined module
	GETPNM	Well-defined module
	LIB$FREE_EF	External routine
	LIB$GET_EF	External routine
	RDMALF	Well-defined module
	SYS$ASSIGN	External routine
	SYS$BINTIM	External routine
	SYS$DASSGN	External routine
	SYS$GETJPI	External routine
	SYS$QIO	External routine
	SYS$QIOW	External routine
	SYS$SETIMR	External routine
	SYS$WAITFR	External routine
	WRTERR	Well-defined module

IPC Main Program Parts List

Main Pgm Name	Module Name	Module Type
RELEVB		Purpose--›RELEASE A TARGET EVENT BLOCK (CLEAR IT) AND
	ERRFTL	Well-defined module
	ERRPRO	Well-defined module
	GETPNM	Well-defined module
	LIB$FREE_EF	External routine
	LIB$GET_EF	External routine
	SDEACH	Well-defined module
	SYS$ASSIGN	External routine
	SYS$BINTIM	External routine
	SYS$DASSGN	External routine
	SYS$GETJPI	External routine
	SYS$QIOW	External routine
	SYS$SETIMR	External routine
	SYS$WAITFR	External routine
	WRTERR	Well-defined module

IPC Main Program Parts List

Main Pgm Name	Module Name	Module Type
SETTIM		Purpose--›START A TIMER
	ERRFTL	Well-defined module
	ERRPRO	Well-defined module
	GETPNM	Well-defined module
	LIB$FREE_EF	External routine
	LIB$GET_EF	External routine
	SETTMR	Well-defined module
	SYS$ASSIGN	External routine
	SYS$BINTIM	External routine
	SYS$DASSGN	External routine
	SYS$GETJPI	External routine
	SYS$QIOW	External routine
	SYS$SETIMR	External routine
	SYS$WAITFR	External routine
	WRTERR	Well-defined module

IPC Main Program Parts List

Main Pgm Name	Module Name	Module Type
SNDMSG		Purpose--›SEND A MESSAGE TO ANOTHER PROGRAM
	ERRFTL	Well-defined module
	ERRPRO	Well-defined module
	GETPNM	Well-defined module
	LIB$FREE_EF	External routine
	LIB$GET_EF	External routine
	SASGCH	Well-defined module
	SYS$ASSIGN	External routine
	SYS$BINTIM	External routine
	SYS$DASSGN	External routine
	SYS$GETJPI	External routine
	SYS$QIOW	External routine
	SYS$SETIMR	External routine
	SYS$WAITFR	External routine
	WRTERR	Well-defined module
	WRTMAI	Well-defined module

IPC Main Program Parts List

Main Pgm Name	Module Name	Module Type
WAIT01		Purpose--›WAIT FOR AN EVENT TO OCCUR
	ERRFTL	Well-defined module
	ERRPRO	Well-defined module
	GETPNM	Well-defined module
	LIB$FREE_EF	External routine
	LIB$GET_EF	External routine
	SWEVTF	Well-defined module
	SYS$ASSIGN	External routine
	SYS$BINTIM	External routine
	SYS$CLREF	External routine
	SYS$DASSGN	External routine
	SYS$GETJPI	External routine
	SYS$QIOW	External routine
	SYS$SETIMR	External routine
	SYS$WAITFR	External routine
	SYS$WFLOR	External routine
	WRTERR	Well-defined module

IPC Main Program Parts List

Main Pgm Name	Module Name	Module Type
WAIT02		Purpose-->WAIT FOR AN EVENT TO OCCUR
	ERRFTL	Well-defined module
	ERRPRO	Well-defined module
	GETPNM	Well-defined module
	LIB$FREE_EF	External routine
	LIB$GET_EF	External routine
	SWEVTF	Well-defined module
	SYS$ASSIGN	External routine
	SYS$BINTIM	External routine
	SYS$CLREF	External routine
	SYS$DASSGN	External routine
	SYS$GETJPI	External routine
	SYS$QIOW	External routine
	SYS$SETIMR	External routine
	SYS$WAITFR	External routine
	SYS$WFLOR	External routine
	WRTERR	Well-defined module

IPC Main Program Parts List

Main Pgm Name	Module Name	Module Type
--------	------	------
WAIT03	Purpose--›WAIT FOR AN EVENT TO OCCUR	
	ERRFTL	Well-defined module
	ERRPRO	Well-defined module
	GETPNM	Well-defined module
	LIB$FREE_EF	External routine
	LIB$GET_EF	External routine
	SWEVTF	Well-defined module
	SYS$ASSIGN	External routine
	SYS$BINTIM	External routine
	SYS$CLREF	External routine
	SYS$DASSGN	External routine
	SYS$GETJPI	External routine
	SYS$QIOW	External routine
	SYS$SETIMR	External routine
	SYS$WAITFR	External routine
	SYS$WFLOR	External routine
	WRTERR	Well-defined module

3.10.8 Module Documentation

The following documentation describes information which is specific to each individual module being documented in this specification as listed in section 3.10.2. It provides a compact way of getting information that would be otherwise buried within each module's source code.

The specific items in this module documentation have the following meanings:

NAME: Name of program Module.

PURPOSE: Purpose of Module as detailed in the source code.

LANGUAGE: Programming language source code is written in.
                               The choices are:
                                                    VAX-11 FORTRAN

        C   (I/S-1 Workbench 'C')
                                                    VAX-11 COBOL

MODULE TYPE: Whether a Program, Subroutine, or Function.

SOURCE FILE: Name of Source File from file specification.

SOURCE FILE TYPE: Source File Extension from file specification.

HOST: Whether this is a host-dependent routine (VAX or IBM) or blank if host-independent.

SUBSYSTEM: IISS sub-system this file resides in.

SUBDIRECTORY: Sub-directory of that subsystem in which this file resides.

DOCUMENTATION GROUP: Name of documentation group of which this source file is a member.

DESCRIPTION: A description of the module as otained

from the source code.

ARGUMENTS: The arguments with which this routine
                             is called if it is a Subroutine or a
                             Function.

INCLUDE FILES: A list of all the files that are
                             included into this module as well as
                             their purposes.

ROUTINES CALLED: Subroutines or Functions, either
                             documented or external, called by
                             this module, if any.

CALLED DIRECTLY BY: The documented routines which call
                             this module, if any.

USED IN MAIN PROGRAM(S):   The documented Main Programs which
                             contain this module in their parts
                             list according to the list in section
                             3.10.7.

The Module Documentation is arranged alphabetically according
to Module Name.

IPC Module Documentation

NAME: CNLTIM
PURPOSE: CANCEL A TIMER
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CNLTIM
SOURCE FILE TYPE: .COB
HOST: VAX
SUBSYSTEM: IPC
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

   CANCEL A RUNNING TIMER

ARGUMENTS:

  TIMER-EVENT-BLOCK = RECRD
  RET-STATUS = DSPLY [X(5)]

INCLUDE FILES:

  CHKSTS   - CHKSTS.INC -- CHECK STATUS
  ERRSTS   - ERRSTS.INC -- IISS ERROR CODES
  TIMREB   - TIMREB.INC -- TIME EVENT BLOCK DESCRIPTION
  ERRPRO   - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

  FREVTF   - FREE THE EVENT FLAG
  CNLTMR   - CANCEL THE TIMER
  ERRPRO   - PROCESS ERROR

IPC Module Documentation

NAME: CNLTMR
PURPOSE: CANCEL THE TIMER
LANGUAGE: VAX-11 FORTRAN
MODULE TYPE: SUBROUTINE
SOURCE FILE: CNLTMR
SOURCE FILE TYPE: .FOR
HOST: VAX
SUBSYSTEM: IPC
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

   CANCEL IISS TIMER
   REQUEST IDENTIFICATION IS 4

ARGUMENTS:

   RSTATS = CHAR
            - RETURN STATUS

INCLUDE FILES:

   ERRSTS.INF - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

   ERRPRO   - PROCESS ERROR
   SYS$CANTIM

CALLED DIRECTLY BY:

   CNLTIM   - CANCEL A TIMER

USED IN MAIN PROGRAM(S):

   CNLTIM   - CANCEL A TIMER

IPC Module Documentation

NAME:	CRTMBX
PURPOSE:	CREATE A MAILBOX
LANGUAGE:	VAX-11 COBOL
MODULE TYPE:	SUBROUTINE
SOURCE FILE:	CRTMBX
SOURCE FILE TYPE:	.COB
HOST:	VAX
SUBSYSTEM:	IPC
SUBDIRECTORY:	
DOCUMENTATION GROUP:	IPC

DESCRIPTION:

   IT CREATES A MAILBOX THROUGH WHICH THE
   PROGRAM WILL RECEIVE MESSAGES FROM ANOTHER
   PROGRAM RUNNING ON THE SAME COMPUTER.

ARGUMENTS:

   INPUT-MAILBOX-NAME = DSPLY [X(14)]
   MAILBOX-SIZE = DSPLY [9(5)]
   MAILBOX-EVENT-BLOCK = RECRD
   RET-STATUS = DSPLY [X(5)]

INCLUDE FILES:

   CHKSTS   - CHKSTS.INC -- CHECK STATUS
   ERRSTS   - ERRSTS.INC -- IISS ERROR CODES
   KIPC   - KIPC.INC -- CONSTAND BLOCK FOR IPC PRIMITIVES
   MBXEBK   - MBXEBK.INC -- MAILBOX EVENT BLOCK DESCRIPTION
   ERRPRO   - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

   VALMBE   - VALIDATE THAT MAILBOX ALREADY EXISTS
   SCRTMB   - CREATE A MAILBOX
   ERRPRO   - PROCESS ERROR

IPC Module Documentation

```
NAME:               DELMBX
PURPOSE:            DELETE A MAILBOX
LANGUAGE:           VAX-11 COBOL
MODULE TYPE:        SUBROUTINE
SOURCE FILE:        DELMBX
SOURCE FILE TYPE:   .COB
HOST:               VAX
SUBSYSTEM:          IPC
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC
```

DESCRIPTION:

   - IT DELETES A MAILBOX WHEREBY REMOVING
   THE ABILITY TO RECEIVE MESSAGES FROM ANOTHER
   PROGRAM THROUGH THE GIVEN INPUT MAILBOX

ARGUMENTS:

   INPUT-MAILBOX-NAME = DSPLY [X(14)]
   MAILBOX-EVENT-BLOCK = RECRD
   RET-STATUS = DSPLY [X(5)]

INCLUDE FILES:

   CHKSTS   - CHKSTS.INC -- CHECK STATUS
   ERRSTS   - ERRSTS.INC -- IISS ERROR CODES
   MBXEBK   - MBXEBK.INC -- MAILBOX EVENT BLOCK DESCRIPTION
   ERRPRO   - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

   SDELMB   - DELETE THE MAILBOX
   SDEACH   - DEASSIGN CHANNEL FROM THE MAILBOX
   ERRPRO   - PROCESS ERROR

IPC Module Documentation

NAME: ENDRUN
PURPOSE: STOP THE PROGRAM
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: ENDRUN
SOURCE FILE TYPE: .COB
HOST: VAX
SUBSYSTEM: IPC
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

   - THE IBM NEEDS A SPECIAL ROUTINE TO
   STOP THE PROGRAM, THEREFORE WE ALL DO.

IPC Module Documentation

NAME: ERRFTL
PURPOSE: PROCESS FATAL ERROR
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ERRFTL
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: ERR
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

  - This module date and time stamps the
  error message, assigns a function code
  and writes the error message to a
  file called ERRFTL

  -

ARGUMENTS:

  RET-STATUS = DSPLY [X(5)]
  MODULE-NAME = DSPLY [X(6)]
  MESG-DESC = DSPLY [X(60)]
  ORIGINAL-ERR-MSG = DSPLY [X(109)]

INCLUDE FILES:

  CHKSTS   - CHKSTS.INC -- CHECK STATUS

ROUTINES CALLED:

  GETPNM   - GET PROCESS NAME   **

CALLED DIRECTLY BY:

  ERRPRO   - PROCESS ERROR

USED IN MAIN PROGRAM(S):

```
CNLTIM    - CANCEL A TIMER
CRTMBX    - CREATE A MAILBOX
DELMBX    - DELETE A MAILBOX
GETMSG    - GET A MESSAGE FROM ANOTHER PROGRAM
LOCKEF    - LOCK EVENT FLAG
RCVMSG    - RECEIVE A MESSAGE FROM ANOTHER PROGRAM
RELEVB    - RELEASE A TARGET EVENT BLOCK (CLEAR IT) AND
SETTIM    - START A TIMER
SNDMSG    - SEND A MESSAGE TO ANOTHER PROGRAM
WAITO1    - WAIT FOR AN EVENT TO OCCUR
WAITO2    - WAIT FOR AN EVENT TO OCCUR
WAITO3    - WAIT FOR AN EVENT TO OCCUR
```

IPC Module Documentation

NAME: ERRPRO
PURPOSE: PROCESS ERROR
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: ERRPRO
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: ERR
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

  LANGUAGE: VAX-11 COBOL

  DESCRIPTION:

ARGUMENTS:

  RET-STATUS = DSPLY [X(5)]
  MODULE-NAME = DSPLY [X(6)]
  MESG-DESC = DSPLY [X(60)]

INCLUDE FILES:

  CHKSTS   - CHKSTS.INC -- CHECK STATUS

ROUTINES CALLED:

  GETPNM   -   GET PROCESS NAME
  WRTERR   -   WRITE MESSAGE TO MAILBOX
  ERRFTL   -   PROCESS FATAL ERROR

CALLED DIRECTLY BY:

  CNLTIM   -   CANCEL A TIMER
  CRTMBX   -   CREATE A MAILBOX
  DELMBX   -   DELETE A MAILBOX

```
GETMSG      - GET A MESSAGE FROM ANOTHER PROGRAM
RCVMSG      - RECEIVE A MESSAGE FROM ANOTHER PROGRAM
RELEVB      - RELEASE A TARGET EVENT BLOCK (CLEAR IT) AND
SETTIM      - START A TIMER
SNDMSG      - SEND A MESSAGE TO ANOTHER PROGRAM
WAITO1      - WAIT FOR AN EVENT TO OCCUR
WAITO2      - WAIT FOR AN EVENT TO OCCUR
WAITO3      - WAIT FOR AN EVENT TO OCCUR
CNLTMR      - CANCEL THE TIMER
FREVTF      - FREE THE EVENT FLAG
LOCKEF      - LOCK EVENT FLAG
RDMALF      - READ THE MAILBOX FOR EVENT FLAG SET
SASGCH      - ASSIGN A CHANNEL TO  A MAILBOX
SCRTMB      - CREATE A MAILBOX
SDEACH      - DEASSIGN CHANNEL FROM THE MAILBOX
SDELMB      - DELETE THE MAILBOX
SETTMR      - SET A TIMER
SWEVTF      - WAIT FOR ONE OF POSSIBLY MANY EVENTS
VALMBE      - VALIDATE THAT MAILBOX ALREADY EXISTS
WRTMAI      - WRITE THE MESSAGE INTO THE MAILBOX
```

USED IN MAIN PROGRAM(S):

```
CNLTIM      - CANCEL A TIMER
CRTMBX      - CREATE A MAILBOX
DELMBX      - DELETE A MAILBOX
GETMSG      - GET A MESSAGE FROM ANOTHER PROGRAM
LOCKEF      - LOCK EVENT FLAG
RCVMSG      - RECEIVE A MESSAGE FROM ANOTHER PROGRAM
RELEVB      - RELEASE A TARGET EVENT BLOCK (CLEAR IT) AND
SETTIM      - START A TIMER
SNDMSG      - SEND A MESSAGE TO ANOTHER PROGRAM
WAITO1      - WAIT FOR AN EVENT TO OCCUR
WAITO2      - WAIT FOR AN EVENT TO OCCUR
WAITO3      - WAIT FOR AN EVENT TO OCCUR
```

IPC Module Documentation

```
NAME:                FREVTF
PURPOSE:             FREE THE EVENT FLAG
LANGUAGE:            VAX-11 FORTRAN
MODULE TYPE:         SUBROUTINE
SOURCE FILE:         FREVTF
SOURCE FILE TYPE:    .FOR
HOST:                VAX
SUBSYSTEM:           IPC
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC
```

DESCRIPTION:

    :- THIS MODULE RELEASES AN EVENT FLAG THAT
   WAS PREVIOUSLY OBTAINED FROM THE SYSTEM.
    :-

ARGUMENTS:

   EVTFLG = I*2
   RSTATS = CHAR

ROUTINES CALLED:

   ERRPRO   - PROCESS ERROR
   LIB$FREE_EF

CALLED DIRECTLY BY:

   CNLTIM   - CANCEL A TIMER
   GETMSG   - GET A MESSAGE FROM ANOTHER PROGRAM

USED IN MAIN PROGRAM(S):

   CNLTIM   - CANCEL A TIMER
   GETMSG   - GET A MESSAGE FROM ANOTHER PROGRAM

IPC Module Documentation

NAME: GETMSG
PURPOSE: GET A MESSAGE FROM ANOTHER PROGRAM
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: GETMSG
SOURCE FILE TYPE: .COB
HOST: VAX
SUBSYSTEM: IPC
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

   - IT ACCEPTS THE MESSAGE THAT WAS SENT FROM
   ANOTHER PORGRAM RUNNING ON THE SAME COMPUTER
   AND MOVES IT INTO THE GIVEN BUFFER.

ARGUMENTS:

   INPUT-MAILBOX-NAME = DSPLY [X(14)]
   BUFFER = DSPLY [X(2000)]
   BUFFER-SIZE = DSPLY [9(4)]
   NUMBER-OF-BYTES = DSPLY [9(4)]
   MAILBOX-EVENT-BLOCK = RECRD
   RET-STATUS = DSPLY [X(5)]

INCLUDE FILES:

   CHKSTS   - CHKSTS.INC -- CHECK STATUS
   KIPC   - KIPC.INC -- CONSTAND BLOCK FOR IPC PRIMITIVES
   ERRSTS   - ERRSTS.INC -- IISS ERROR CODES
   MBXEBK   - MBXEBK.INC -- MAILBOX EVENT BLOCK DESCRIPTION
   ERRPRO   - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

   FREVTF   - FREE THE EVENT FLAG
   ERRPRO   - PROCESS ERROR

IPC Module Documentation

NAME: GETPNM
PURPOSE: GET PROCESS NAME
LANGUAGE: VAX-11 FORTRAN
MODULE TYPE: SUBROUTINE
SOURCE FILE: GETPNM
SOURCE FILE TYPE: .FOR
HOST:
SUBSYSTEM: ERR
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

   GET THE NAME OF THE CALLING PROCESS

ARGUMENTS:

   PRCNAM = CHAR
             - Process name
   RSTATS = CHAR
             - Return status

INCLUDE FILES:

   ERRSTS.INF - **** PURPOSE NOT FOUND BY STRIPPER ****
   ($JPIDEF) - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

   SYS$GETJPI

CALLED DIRECTLY BY:

   ERRFTL   - PROCESS FATAL ERROR
   ERRPRO   - PROCESS ERROR

USED IN MAIN PROGRAM(S):

CNLTIM   - CANCEL A TIMER
CRTMBX   - CREATE A MAILBOX
DELMBX   - DELETE A MAILBOX
GETMSG   - GET A MESSAGE FROM ANOTHER PROGRAM
LOCKEF   - LOCK EVENT FLAG
RCVMSG   - RECEIVE A MESSAGE FROM ANOTHER PROGRAM
RELEVB   - RELEASE A TARGET EVENT BLOCK (CLEAR IT) AND
SETTIM   - START A TIMER
SNDMSG   - SEND A MESSAGE TO ANOTHER PROGRAM
WAITO1   - WAIT FOR AN EVENT TO OCCUR
WAITO2   - WAIT FOR AN EVENT TO OCCUR
WAITO3   - WAIT FOR AN EVENT TO OCCUR

IPC Module Documentation

NAME: LOCKEF
PURPOSE: LOCK EVENT FLAG
LANGUAGE: VAX-11 FORTRAN
MODULE TYPE: SUBROUTINE
SOURCE FILE: LOCKEF
SOURCE FILE TYPE: .FOR
HOST: VAX
SUBSYSTEM: IPC
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

    :- LOCK EVENT FLAG FROM BEING USED UNTIL FREED.

INCLUDE FILES:

   ERRSTS.INF - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

   ERRPRO   - PROCESS ERROR
   LIB$GET_EF

IPC Module Documentation

NAME: RCVMSG
PURPOSE: RECEIVE A MESSAGE FROM ANOTHER PROGRAM
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: RCVMSG
SOURCE FILE TYPE: .COB
HOST: VAX
SUBSYSTEM: IPC
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

ARGUMENTS:

   INPUT-MAILBOX-NAME = DSPLY [X(14)]
   EVENT-NUMBER = DSPLY [99]
   MAILBOX-EVENT-BLOCK = RECRD
   RET-STATUS = DSPLY [X(5)]

INCLUDE FILES:

   CHKSTS   - CHKSTS.INC -- CHECK STATUS
   ERRSTS   - ERRSTS.INC -- IISS ERROR CODES
   KIPC   - KIPC.INC -- CONSTAND BLOCK FOR IPC PRIMITIVES
   MBXEBK   - MBXEBK.INC -- MAILBOX EVENT BLOCK DESCRIPTION
   ERRPRO   - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

   RDMALF   - READ THE MAILBOX FOR EVENT FLAG SET
   ERRPRO   - PROCESS ERROR

IPC Module Documentation

NAME: RDMALF
PURPOSE: READ THE MAILBOX FOR EVENT FLAG SET
LANGUAGE: VAX-11 FORTRAN
MODULE TYPE: SUBROUTINE
SOURCE FILE: RDMALF
SOURCE FILE TYPE: .FOR
HOST: VAX
SUBSYSTEM: IPC
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

   :- ISSUE QIO WITH EVENT FLAG TO READ MESSAGE
  FROM MAILBOX USING THE CHANNEL NUMBER

ARGUMENTS:

  CHANNL = I*2
           -CHANNEL NUMBER
  EVTFLG = I*2
           -EVENT FLAG
  EVTBLK = L*1 (1)
           -MAILBOX EVENT BLOCK
  BUFLEN = I*2
           -MAXIMUM BUFFER LENGTH
  RSTATS = CHAR
           -RETURN STATUS

INCLUDE FILES:

  ERRSTS.INF - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

  ERRPRO   - PROCESS ERROR
  LIB$GET_EF
  SYS$QIO

CALLED DIRECTLY BY:

  RCVMSG   - RECEIVE A MESSAGE FROM ANOTHER PROGRAM

USED IN MAIN PROGRAM(S):

  RCVMSG   - RECEIVE A MESSAGE FROM ANOTHER PROGRAM

IPC Module Documentation

NAME: RELEVB
PURPOSE: RELEASE A TARGET EVENT BLOCK (CLEAR IT)
                         AND
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: RELEVB
SOURCE FILE TYPE: .COB
HOST: VAX
SUBSYSTEM: IPC
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

  - DEASSIGN THE CHANNEL ASSIGNED TO THE
  TARGET MAILBOX.

ARGUMENTS:

  TARGET-MAILBOX-NAME = DSPLY [X(14)]
  MAILBOX-EVENT-BLOCK = RECRD
  RET-STATUS = DSPLY [X(5)]

INCLUDE FILES:

  ERRSTS   - ERRSTS.INC -- IISS ERROR CODES
  CHKSTS   - CHKSTS.INC -- CHECK STATUS
  MBXEBK   - MBXEBK.INC -- MAILBOX EVENT BLOCK DESCRIPTION
  ERRPRO   - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

  SDEACH   - DEASSIGN CHANNEL FROM THE MAILBOX
  ERRPRO   - PROCESS ERROR

IPC Module Documentation


```
NAME:                   SASGCH
PURPOSE:                ASSIGN A CHANNEL TO  A MAILBOX
LANGUAGE:               VAX-11 FORTRAN
MODULE TYPE:            SUBROUTINE
SOURCE FILE:            SASGCH
SOURCE FILE TYPE:       .FOR
HOST:                   VAX
SUBSYSTEM:              IPC
SUBDIRECTORY:
DOCUMENTATION GROUP:    IPC
```

DESCRIPTION:

   :-ASSIGN A CHANNEL NUMBER TO MAILBOX

ARGUMENTS:

  MBXNAM = CHAR
       -MAILBOX NAME
  CHANNL = I*2
       -CHANNEL NUMBER
  RSTATS = CHAR
       -RETURN STATUS

INCLUDE FILES:

  ERRSTS.INF - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

  ERRPRO   - PROCESS ERROR
  SYS$ASSIGN

CALLED DIRECTLY BY:

  SNDMSG   - SEND A MESSAGE TO ANOTHER PROGRAM

USED IN MAIN PROGRAM(S):

SNDMSG   - SEND A MESSAGE TO ANOTHER PROGRAM

IPC Module Documentation

NAME: SCRTMB
PURPOSE: CREATE A MAILBOX
LANGUAGE: VAX-11 FORTRAN
MODULE TYPE: SUBROUTINE
SOURCE FILE: SCRTMB
SOURCE FILE TYPE: .FOR
HOST: VAX
SUBSYSTEM: IPC
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

    :- TO CREATE A MAILBOX USING THE NAME AND THE
  SIZE SPECIFIED. IT ALSO DISABLES THE RESOURCE
  WAIT MODE TO ALLOW CONTROL RETURN TO CALLING PROGRAM
  IMMEDIATELY WITHOUT WAITING FOR THE RESOURCE
  TO BE AVAILABLE, SUCH AS THE MAILBOX FULL.

ARGUMENTS:

  MBXNAM = CHAR
            -MAILBOX NAME
  MBXSIZ = I*2
            -MAILBOX SIZE
  CHANNL = I*2
            -CHANNEL NUMBER
  RSTATS = CHAR
            -RETURN STATUS

INCLUDE FILES:

  ERRSTS.INF - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

  ERRPRO   - PROCESS ERROR
  SYS$CREMBX
  SYS$SETRWM

CALLED DIRECTLY BY:

   CRTMBX   - CREATE A MAILBOX

USED IN MAIN PROGRAM(S):

   CRTMBX   - CREATE A MAILBOX

IPC Module Documentation

NAME: SDEACH
PURPOSE: DEASSIGN CHANNEL FROM THE MAILBOX
LANGUAGE: VAX-11 FORTRAN
MODULE TYPE: SUBROUTINE
SOURCE FILE: SDEACH
SOURCE FILE TYPE: .FOR
HOST: VAX
SUBSYSTEM: IPC
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

    :-DEASSIGN A MAILBOX CHANNEL NUMBER

ARGUMENTS:

  CHANNL = I*2
            -CHANNEL NUMBER
  RSTATS = CHAR
            -RETURN STATUS

INCLUDE FILES:

  ERRSTS.INF - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

  ERRPRO   - PROCESS ERROR
  SYS$DASSGN

CALLED DIRECTLY BY:

  DELMBX   - DELETE A MAILBOX
  RELEVB   - RELEASE A TARGET EVENT BLOCK (CLEAR IT) AND

USED IN MAIN PROGRAM(S):

DELMBX   - DELETE A MAILBOX
RELEVB   - RELEASE A TARGET EVENT BLOCK (CLEAR IT) AND

IPC Module Documentation

NAME: SDELMB
PURPOSE: DELETE THE MAILBOX
LANGUAGE: VAX-11 FORTRAN
MODULE TYPE: SUBROUTINE
SOURCE FILE: SDELMB
SOURCE FILE TYPE: .FOR
HOST: VAX
SUBSYSTEM: IPC
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

    :- TO DELETE A MAILBOX USING THE CHANNEL NUMBER

ARGUMENTS:

   CHANNL =   I*2
              -CHANNEL NUMBER
   RSTATS =   CHAR
              -RETURN STATUS

INCLUDE FILES:

   ERRSTS.INF - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

   ERRPRO   - PROCESS ERROR
   SYS$DELMBX

CALLED DIRECTLY BY:

   DELMBX   - DELETE A MAILBOX

USED IN MAIN PROGRAM(S):

   DELMBX   - DELETE A MAILBOX

IPC Module Documentation

NAME: SETTIM
PURPOSE: START A TIMER
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: SETTIM
SOURCE FILE TYPE: .COB
HOST: VAX
SUBSYSTEM: IPC
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

   REQUEST A TIMER TO START RUNNING. THE
   PROGRAM CONTINUES TO EXECUTE WITH THE
   ELAPSED TIME BEING DISCOVERED IN THE
   WAITEB PRIMITIVE.

ARGUMENTS:

   TIME-INTERVAL = RECRD
   EVENT-NUMBER = DSPLY [99]
   TIMER-EVENT-BLOCK = RECRD
   RET-STATUS = DSPLY [X(5)]

INCLUDE FILES:

   CHKSTS   - CHKSTS.INC -- CHECK STATUS
   ERRSTS   - ERRSTS.INC -- IISS ERROR CODES
   TIMREB   - TIMREB.INC -- TIME EVENT BLOCK DESCRIPTION
   ERRPRO   - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

   SETTMR   - SET A TIMER
   ERRPRO   - PROCESS ERROR

IPC Module Documentation

NAME: SETTMR
PURPOSE: SET A TIMER
LANGUAGE: VAX-11 FORTRAN
MODULE TYPE: SUBROUTINE
SOURCE FILE: SETTMR
SOURCE FILE TYPE: .FOR
HOST: VAX
SUBSYSTEM: IPC
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

    SET A TIMER WITH AN EVENT FLAG
    REQUEST IDENTIFICATION IS 4

ARGUMENTS:

    INTVAL = CHAR
             - TIME INTERVAL - HHMMSS
    EVTFLG = I*2
             - EVENT FLAG NUMBER
    RSTATS = CHAR
             - RETURN STATUS

INCLUDE FILES:

    ERRSTS.INF - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

    ERRPRO   - PROCESS ERROR
    LIB$GET_EF
    SYS$BINTIM
    SYS$SETIMR

CALLED DIRECTLY BY:

    SETTIM   - START A TIMER

USED IN MAIN PROGRAM(S):

  SETTIM   - START A TIMER

IPC Module Documentation

NAME: SNDMSG
PURPOSE: SEND A MESSAGE TO ANOTHER PROGRAM
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: SNDMSG
SOURCE FILE TYPE: .COB
HOST: VAX
SUBSYSTEM: IPC
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

  - IT SENDS A MESSAGE TO ANOTHER PROGRAM
  RUNNING ON THE SAME COMPUTER THROUGH THE
  INPUT MAILBOX OF THE OTHER PROGRAM.

ARGUMENTS:

  TARGET-MAILBOX-NAME = DSPLY [X(14)]
  BUFFER = DSPLY [X(2000)]
  NUMBER-OF-BYTES = DSPLY [9(4)]
  MAILBOX-EVENT-BLOCK = RECRD
  RET-STATUS = DSPLY [X(5)]

INCLUDE FILES:

    CHKSTS   - CHKSTS.INC -- CHECK STATUS
    ERRSTS   - ERRSTS.INC -- IISS ERROR CODES
    KIPC   - KIPC.INC -- CONSTAND BLOCK FOR IPC PRIMITIVES
    MBXEBK   - MBXEBK.INC -- MAILBOX EVENT BLOCK DESCRIPTION
    ERRPRO   - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

    WRTMAI   - WRITE THE MESSAGE INTO THE MAILBOX
    SASGCH   - ASSIGN A CHANNEL TO A MAILBOX
    ERRPRO   - PROCESS ERROR

IPC Module Documentation

NAME: SWEVTF
PURPOSE: WAIT FOR ONE OF POSSIBLY MANY EVENTS
LANGUAGE: VAX-11 FORTRAN
MODULE TYPE: SUBROUTINE
SOURCE FILE: SWEVTF
SOURCE FILE TYPE: .FOR
HOST: VAX
SUBSYSTEM: IPC
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

   :- THIS MODULE SET UP THE EVENT CLUSTER AND WAITS
   FOR ONE OF THE EVENT TO HAPPEN.
   :-

ARGUMENTS:

   ENARRY =   I*2   (22)
   EVTFLG =   I*2   (22)
   EVTNUM =   I*2
   RSTATS =   CHAR

ROUTINES CALLED:

   ERRPRO   - PROCESS ERROR
   LIB$FREE_EF
   SYS$CLREF
   SYS$WFLOR

CALLED DIRECTLY BY:

   WAITO1   - WAIT FOR AN EVENT TO OCCUR
   WAITO2   - WAIT FOR AN EVENT TO OCCUR
   WAITO3   - WAIT FOR AN EVENT TO OCCUR

USED IN MAIN PROGRAM(S):

   WAITO1   - WAIT FOR AN EVENT TO OCCUR

WAIT02   - WAIT FOR AN EVENT TO OCCUR
WAIT03   - WAIT FOR AN EVENT TO OCCUR

IPC Module Documentation

NAME: VALMBE
PURPOSE: VALIDATE THAT MAILBOX ALREADY EXISTS
LANGUAGE: VAX-11 FORTRAN
MODULE TYPE: SUBROUTINE
SOURCE FILE: VALMBE
SOURCE FILE TYPE: .FOR
HOST: VAX
SUBSYSTEM: IPC
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

   :- THIS MODULE CHECKS TO SEE IF MAILBOX
ALREADY EXISTS.

ARGUMENTS:

  INMBNM = CHAR
      -INPUT MAILBOX NAME
  RSTATS = CHAR
      -RET STATUS

INCLUDE FILES:

  ERRSTS.INF - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

  ERRPRO   - PROCESS ERROR
  SYS$GETDEV

CALLED DIRECTLY BY:

  CRTMBX   - CREATE A MAILBOX

USED IN MAIN PROGRAM(S):

CRTMBX   - CREATE A MAILBOX

IPC Module Documentation

```
NAME:                  WAITO1
PURPOSE:               WAIT FOR AN EVENT TO OCCUR
LANGUAGE:              VAX-11 COBOL
MODULE TYPE:           SUBROUTINE
SOURCE FILE:           WAITO1
SOURCE FILE TYPE:      .COB
HOST:                  VAX
SUBSYSTEM:             IPC
SUBDIRECTORY:
DOCUMENTATION GROUP:   IPC
```

DESCRIPTION:

   - IT WAITS FOR ONE OF OUTSTANDING REQUESTS THAT
   ARE ASSOCIATED WITH THE LIST OF EVENT BLOCKS
   TO BE SATISFIED. THE EVENT NUMBER ASSOCIATED
   WITH THE COMPLETED REQUEST IS RETURNED IN THE
   EVENT NUMBER VARIABLE.

ARGUMENTS:

   EVENT-NUMBER = DSPLY [99]
   RET-STATUS = DSPLY [X(5)]
   NUMBER-OF-EVENT-BLOCKS = DSPLY [99]
   MBX-EVENT-BLOCK-01 = RECRD

INCLUDE FILES:

   CHKSTS   - CHKSTS.INC -- CHECK STATUS
   ERRSTS   - ERRSTS.INC -- IISS ERROR CODES
   MBEBO1   - MBEBO1.INC -- MAILBOX EVENT BLOCK DESCRIPTION
   ERRPRO   - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

   SWEVTF   - WAIT FOR ONE OF POSSIBLY MANY EVENTS
   ERRPRO   - PROCESS ERROR

IPC Module Documentation

NAME: WAIT02
PURPOSE: WAIT FOR AN EVENT TO OCCUR
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: WAIT02
SOURCE FILE TYPE: .COB
HOST: VAX
SUBSYSTEM: IPC
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

   - IT WAITS FOR ONE OF OUTSTANDING REQUESTS THAT
   ARE ASSOCIATED WITH THE LIST OF EVENT BLOCKS
   TO BE SATISFIED. THE EVENT NUMBER ASSOCIATED
   WITH THE COMPLETED REQUEST IS RETURNED IN THE
   EVENT NUMBER VARIABLE.

ARGUMENTS:

   EVENT-NUMBER = DSPLY [99]
   RET-STATUS = DSPLY [X(5)]
   NUMBER-OF-EVENT-BLOCKS = DSPLY [99]
   MBX-EVENT-BLOCK-01 = RECRD
   MBX-EVENT-BLOCK-02 = RECRD

INCLUDE FILES:

   CHKSTS   - CHKSTS.INC -- CHECK STATUS
   ERRSTS   - ERRSTS.INC -- IISS ERROR CODES
   MBEB01   - MBEB01.INC -- MAILBOX EVENT BLOCK DESCRIPTION
   MBEB02   - MBEB02.INC -- MAILBOX EVENT BLOCK DESCRIPTION
   ERRPRO   - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

   SWEVTF   - WAIT FOR ONE OF POSSIBLY MANY EVENTS

ERRPRO   - PROCESS ERROR

IPC Module Documentation

NAME:	WAITO3
PURPOSE:	WAIT FOR AN EVENT TO OCCUR
LANGUAGE:	VAX-11 COBOL
MODULE TYPE:	SUBROUTINE
SOURCE FILE:	WAITO3
SOURCE FILE TYPE:	.COB
HOST:	VAX
SUBSYSTEM:	IPC
SUBDIRECTORY:	
DOCUMENTATION GROUP:	IPC

DESCRIPTION:

- IT WAITS FOR ONE OF OUTSTANDING REQUESTS THAT
ARE ASSOCIATED WITH THE LIST OF EVENT BLOCKS
TO BE SATISFIED. THE EVENT NUMBER ASSOCIATED
WITH THE COMPLETED REQUEST IS RETURNED IN THE
EVENT NUMBER VARIABLE.

ARGUMENTS:

  EVENT-NUMBER = DSPLY [99]
  RET-STATUS = DSPLY [X(5)]
  NUMBER-OF-EVENT-BLOCKS = DSPLY [99]
  MBX-EVENT-BLOCK-01 = RECRD
  MBX-EVENT-BLOCK-02 = RECRD
  MBX-EVENT-BLOCK-03 = RECRD

INCLUDE FILES:

  CHKSTS   - CHKSTS.INC -- CHECK STATUS
  ERRSTS   - ERRSTS.INC -- IISS ERROR CODES
  MBEB01   - MBEB01.INC -- MAILBOX EVENT BLOCK DESCRIPTION
  MBEB02   - MBEB02.INC -- MAILBOX EVENT BLOCK DESCRIPTION
  MBEB03   - MBEB03.INC -- MAILBOX EVENT BLOCK DESCRIPTION
  ERRPRO   - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

SWEVTF   - WAIT FOR ONE OF POSSIBLY MANY EVENTS
ERRPRO   - PROCESS ERROR

IPC Module Documentation

```
NAME:                 WRTERR
PURPOSE:               WRITE MESSAGE TO MAILBOX
LANGUAGE:             VAX-11 FORTRAN
MODULE TYPE:          SUBROUTINE
SOURCE FILE:          WRTERR
SOURCE FILE TYPE:     .FOR
HOST:
SUBSYSTEM:            ERR
SUBDIRECTORY:
DOCUMENTATION GROUP:  IPC
```

DESCRIPTION:

:- ISSUE QIO TO WRITE A MESSAGE TO MAILBOX

ARGUMENTS:

```
  MBXNAM =  CHAR
            -MAILBOX NAME
  BUFLEN =  I*2
            -MAXIMUM BUFFER LENGTH
  INBUF =  L*1  (1)
            -MESSAGE
  RSTATS =  CHAR
            -RETURN STATUS
```


INCLUDE FILES:

ERRSTS.INF - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

```
  LIB$FREE_EF
  LIB$GET_EF
  SYS$ASSIGN
  SYS$BINTIM
  SYS$DASSGN
  SYS$QIOW
  SYS$SETIMR
  SYS$WAITFR
```

CALLED DIRECTLY BY:

  ERRPRO   - PROCESS ERROR

USED IN MAIN PROGRAM(S):

    CNLTIM   - CANCEL A TIMER
    CRTMBX   - CREATE A MAILBOX
    DELMBX   - DELETE A MAILBOX
    GETMSG   - GET A MESSAGE FROM ANOTHER PROGRAM
    LOCKEF   - LOCK EVENT FLAG
    RCVMSG   - RECEIVE A MESSAGE FROM ANOTHER PROGRAM
    RELEVB   - RELEASE A TARGET EVENT BLOCK (CLEAR IT) AND
    SETTIM   - START A TIMER
    SNDMSG   - SEND A MESSAGE TO ANOTHER PROGRAM
    WAITO1   - WAIT FOR AN EVENT TO OCCUR
    WAITO2   - WAIT FOR AN EVENT TO OCCUR
    WAITO3   - WAIT FOR AN EVENT TO OCCUR

IPC Module Documentation

NAME: WRTMAI
PURPOSE: WRITE THE MESSAGE INTO THE MAILBOX
LANGUAGE: VAX-11 FORTRAN
MODULE TYPE: SUBROUTINE
SOURCE FILE: WRTMAI
SOURCE FILE TYPE: .FOR
HOST: VAX
SUBSYSTEM: IPC
SUBDIRECTORY:
DOCUMENTATION GROUP: IPC

DESCRIPTION:

    :- ISSUE QIO TO WRITE A MESSAGE TO MAILBOX

ARGUMENTS:

  CHANNL = I*2
            -CHANNEL NUMBER
  BUFLEN = I*2
            -MAXIMUM BUFFER LENGTH
  INBUF = L*1 (1)
            -MESSAGE
  RSTATS = CHAR
            -RETURN STATUS

INCLUDE FILES:

  ERRSTS.INF - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

  ERRPRO   - PROCESS ERROR
  SYS$QIOW

CALLED DIRECTLY BY:

  SNDMSG   - SEND A MESSAGE TO ANOTHER PROGRAM

USED IN MAIN PROGRAM(S):

  SNDMSG   - SEND A MESSAGE TO ANOTHER PROGRAM

3.10.9   Include File Descriptions

The following list contains a purpose and description of each include file listed in 3.10.4 as specified in the source code.   The language it is written in is also given.

IPC Include File Description

FILE NAME: CHKSTS
PURPOSE: CHKSTS.INC -- CHECK STATUS
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

IPC Include File Description

FILE NAME: ERRPRO
PURPOSE: PROCESS ERROR INCLUDE FILE
LANGUAGE: VAX-11 COBOL
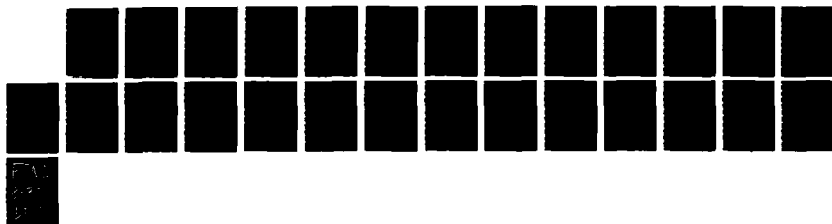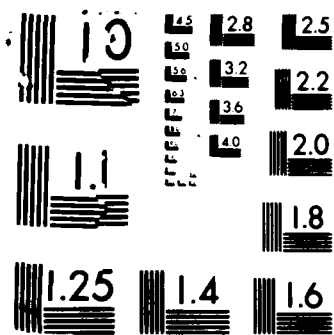
DESCRIPTION:

IPC Include File Description

FILE NAME: ERRSTS
PURPOSE: ERRSTS.INC -- IISS ERROR CODES
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

     THIS FILE CONTAINS ALL IISS ERROR CODES DEFINED IN   *
   COBOL FORMAT   *

IPC Include File Description

FILE NAME: KIPC
PURPOSE: KIPC.INC -- CONSTAND BLOCK FOR IPC PRIMITIVES
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

    THIS FILE CONTAINS ALL CONSTANT DATA ITEMS   *
    USED IN THE IPC PRIMITIVES.   *

IPC Include File Description

FILE NAME: MBEB01
PURPOSE: MBEB01.INC -- MAILBOX EVENT BLOCK DESCRIPTION
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

IPC Include File Description

FILE NAME: MBEB02
PURPOSE: MBEB02.INC -- MAILBOX EVENT BLOCK DESCRIPTION
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

IPC Include File Description

FILE NAME: MBEB03
PURPOSE: MBEB03.INC -- MAILBOX EVENT BLOCK DESCRIPTION
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

IPC Include File Description

FILE NAME: MBXEBK
PURPOSE: MBXEBK.INC -- MAILBOX EVENT BLOCK DESCRIPTION
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

IPC Include File Description

FILE NAME: TIMREB
PURPOSE: TIMREB.INC -- TIME EVENT BLOCK DESCRIPTION
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

3.10.10 Hierarchy Chart

     The following hierarchy charts show the relationships
between all of the modules mentioned in the above documentation.
A module may call a subroutine several times within its code,
but the call will only be shown once as a single relationship on
this hierarchy chart. All modules shown at the top of the first
page are considered Main Programs as described in section 3.10.1
above.

     There is an internal paging scheme as marked by the numbers
in the upper right corner of each page. An index after the last
page of the chart shows where a routine and its calls are first
defined. If a routine has no page reference, it either makes no
calls or is an external routine. A continuation box on the end
of a tree limb shows where that the tree continues on the page
numbered mentioned. A number in a box with a routine name
points to the page where the routine is further defined within
the hierarchy tree. If there is no number in a box, the routine
either makes no calls or is an external routine.

```
                                                                     1
+------+    +------+    +------+    +------+    +------+    +------+
|CNLTIM|    |CRTMBX|    |DELMBX|    |GETMSG|    |LOCKEF|    |(CONT)|
+-----2+    +-----3+    +-----4+    +--+---+    +-----5+    +-----6+
                                       |
                                    +----+----+
                                    |         |
                                 +--+---+   +--+---+
                                 |FREVTF|   |ERRPRO|
                                 +-----2+   +-----7+
```

```
                              +------+
                              |CNLTIM|
                              +--+---+
                                 |
             +--------------------+--------+----------+
             |                             |          |
         +--+---+                      +--+---+   +--+---+
         |FREVTF|                      |CNLTMR|   |ERRPRO|
         +--+---+                      +--+---+   +-----7+
            |                             |
      +-----+------+                +-----+-----+
      |            |                |           |
   +--+---+  +-----+-----+      +--+---+   +----+-----+
   |ERRPRO|  |LIB$FREE_EF|      |ERRPRO|   |SYS$CANTIM|
   +-----7+  +-----------+      +-----7+   +----------+
```

3

```
                        +------+
                        |CRTMBX|
                        +--+---+
                           |
          +----------------+----------+---------+
          |                           |         |
       +--+---+                    +--+---+  +--+---+
       |VALMBE|                    |SCRTMB|  |ERRPRO|
       +--+---+                    +--+---+  +-----7+
          |                           |
    +-----+-----+            +--------++----------+
    |           |            |         |          |
 +--+---+  +----+-----+  +--+---+  +----+-----+  +----+-----+
 |ERRPRO|  |SYS$GETDEV|  |ERRPRO|  |SYS$CREMBX|  |SYS$SETRWM|
 +-----7+  +----------+  +-----7+  +----------+  +----------+
```

4

```
                            +------+
                            |DELMBX|
                            +--+---+
                               |
              +----------------+----------+---------+
              |                           |         |
          +--+---+                    +--+---+   +--+---+
          |SDELMB|                    |SDEACH|   |ERRPRO|
          +--+---+                    +--+---+   +-----7+
             |                           |
      +------+-----+              +------+-----+
      |            |              |            |
  +--+---+   +----+-----+    +--+---+   +----+-----+
  |ERRPRO|   |SYS$DELMBX|    |ERRPRO|   |SYS$DASSGN|
  +-----7+   +----------+    +-----7+   +----------+
```
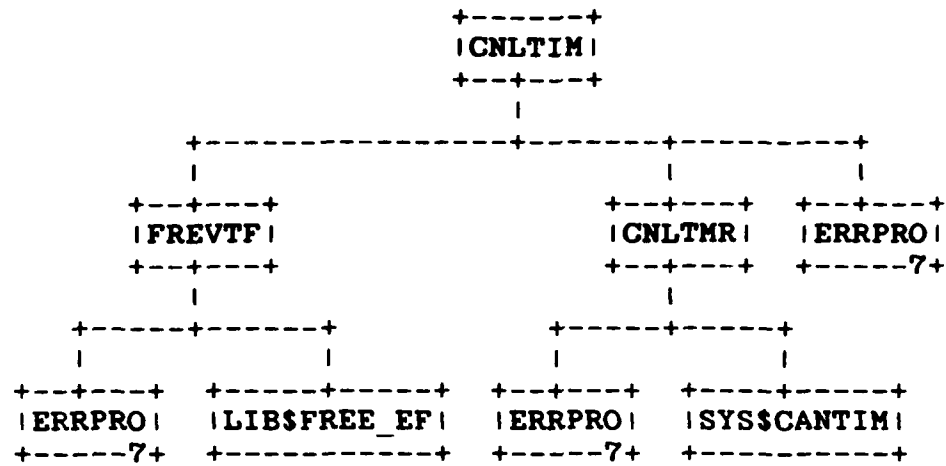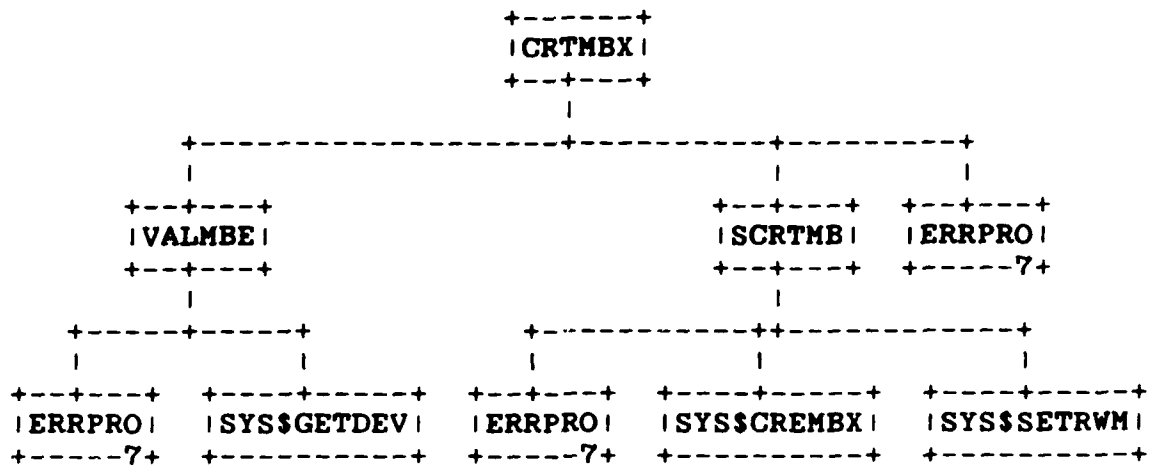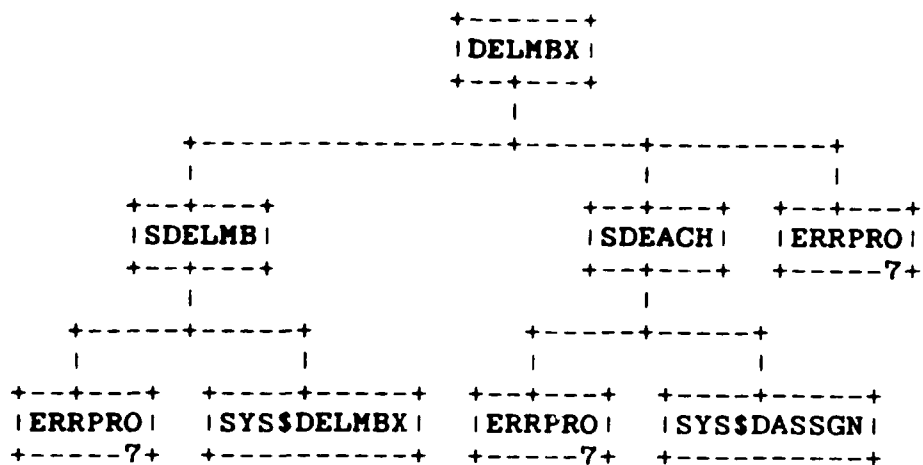
5

```
              +------+
              |LOCKEF|
              +--+---+
                 |
        +-----+-----+
        |           |
    +--+---+    +----+-----+
    |ERRPRO|    |LIB$GET_EF|
    +-----7+    +----------+
```

6

```
+------+   +------+   +------+   +------+   +------+   +------+
|(CONT)|   |RCVMSG|   |RELEVB|   |SETTIM|   |SNDMSG|   |(CONT)|
+-----1+   +-----8+   +--+---+   +-----9+   +----10+   +----11+
                         |
                      +----+----+
                      |         |
                   +--+---+  +--+---+
                   |SDEACH|  |ERRPRO|
                   +-----4+  +-----7+
```

7

```
                        +------+
                        |ERRPRO|
                        +--+---+
                           |
            +-------------+---------+
            |             |         |
        +--+---+      +--+---+   +--+---+
        |GETPNM|      |WRTERR|   |ERRFTL|
        +--+---+      +-----12+  +--+---+
           |                        |
           |                        |
           |                        |
   +----+-----+              +--+---+
   |SYS$GETJPI|              |GETPNM|
   +----------+              +-----7+
```

8

```
                      +------+
                      |RCVMSG|
                      +--+---+
                         |
                      +----+----+
                      |         |
                  +--+---+   +--+---+
                  |RDMALF|   |ERRPRO|
                  +--+---+   +-----7+
                     |
        +------------+------------+
        |            |            |
    +--+---+   +----+-----+   +---+---+
    |ERRPRO|   |LIB$GET_EF|   |SYS$QIO|
    +-----7+   +----------+   +-------+
```

```
                        +------+
                        |SETTIM|
                        +--+---+
                           |
                      +----+----+
                      |         |
                  +--+---+  +--+---+
                  |SETTMR|  |ERRPRO|
                  +--+---+  +-----7+
                     |
     +-----------+--------+-----+------------+
     |           |        |                  |
  +--+---+  +----+-----+  +----+-----+  +----+-----+
  |ERRPRO|  |LIB$GET_EF|  |SYS$BINTIM|  |SYS$SETIMR|
  +-----7+  +----------+  +----------+  +----------+
```
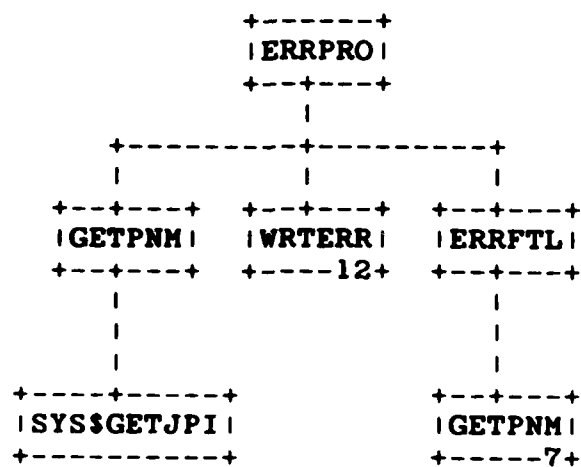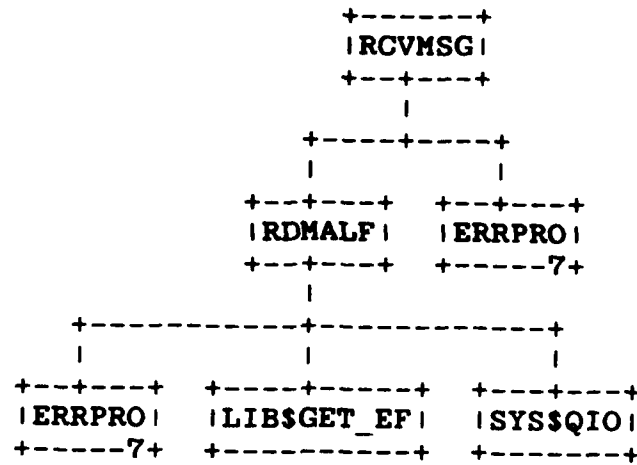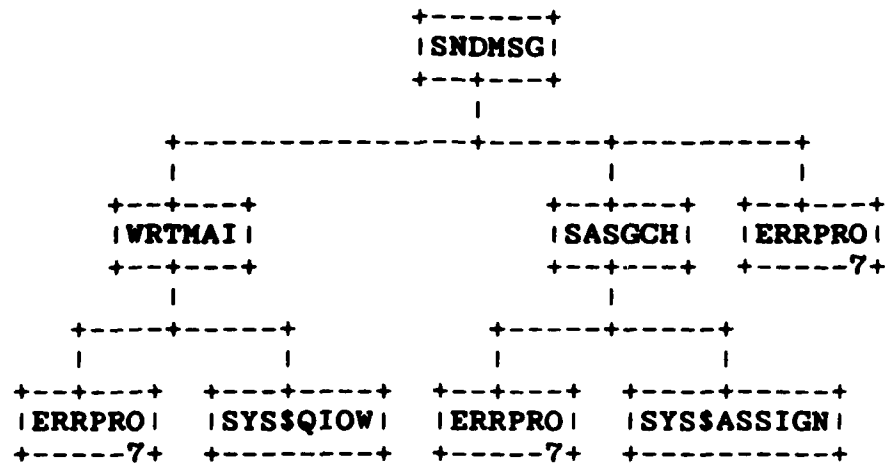
10

```
                          +------+
                          |SNDMSG|
                          +--+---+
                             |
          +--------------------+-------+---------+
          |                            |         |
       +--+---+                     +--+---+   +--+---+
       |WRTMAI|                     |SASGCH|   |ERRPRO|
       +--+---+                     +--+---+   +-----7+
          |                            |
     +----+-----+               +-----+-----+
     |          |               |           |
  +--+---+  +---+----+      +--+---+   +----+-----+
  |ERRPRO|  |SYS$QIOW|      |ERRPRO|   |SYS$ASSIGN|
  +-----7+  +--------+      +-----7+   +----------+
```

11

```
+------+   +------+   +------+                    +------+
|(CONT)|   |WAIT01|   |WAIT02|                    |WAIT03|
+-----6+   +----13+   +--+---+                    +--+---+
                         |                           |
                      +----+----+                 +----+----+
                      |         |                 |         |
                   +--+---+  +--+---+          +--+---+  +--+---+
                   |SWEVTF|  |ERRPRO|          |SWEVTF|  |ERRPRO|
                   +----13+  +-----7+          +----13+  +-----7+
```

12

```
                          +------+
                          |WRTERR|
                          +--+---+
                             |
       +-------------+-----+-------+----------+
       |             |             |          |
  +-----+-----+  +----+-----+  +----+-----+  +--+---+
  |LIB$FREE_EF|  |LIB$GET_EF|  |SYS$ASSIGN|  |(CONT)|
  +-----------+  +----------+  +----------+  +----14+
```

13

```
                        +------+
                        |WAIT01|
                        +--+---+
                           |
                     +----+----+
                     |         |
                 +--+---+  +--+---+
                 |SWEVTF|  |ERRPRO|
                 +--+---+  +-----7+
                    |
    +-----------+--------+------+------------+
    |           |        |                   |
+--+---+  +-----+-----+  +----+----+  +----+-----+
|ERRPRO|  |LIB$FREE_EF|  |SYS$CLREF|  |SYS$WFLOR|
+-----7+  +-----------+  +---------+  +---------+
```

14

```
                          +------+
                          |WRTERR|
                          +--+---+
                             |
      +------------+-----------++-----------+-----------+
      |            |            |           |           |
   +--+---+   +----+-----+   +----+-----+   +---+----+   +--+---+
   |(CONT)|   |SYS$BINTIM|   |SYS$DASSGN|   |SYS$QIOW|   |(CONT)|
   +----12+   +----------+   +----------+   +--------+   +----15+
```

15

```
                    +------+
                    |WRTERR|
                    +--+---+
                       |
        +-----------++------------+
        |           |             |
     +--+---+   +----+-----+   +----+-----+
     |(CONT)|   |SYS$SETIMR|   |SYS$WAITFR|
     +----14+   +----------+   +----------+
```

3.11 Program Listings Comments

This information is contained in the Module Descriptions in section 3.10.

SECTION 4

QUALITY ASSURANCE PROVISIONS

4.1 Introduction and Definitions

"Testing" is a systematic process that may be preplanned and explicitly stated. Test techniques and procedures may be defined in advance, and a sequence of test steps may be specified. "Debugging" is the process of isolation and correction of the cause of an error.

"Antibugging" is defined as the philosophy of writing programs in such a way as to make bugs less likely to occur and when they do occur, to make them more noticeable to the programmer and the user. In other words, as much error checking as is practical and possible in each routine should be performed.

4.2 Computer Programming Test and Evaluation

The quality assurance provisions for test consists of the normal testing techniques that are accomplished during the construction process. They consist of design and code walk-throughs, unit testing, and integration testing. These tests are performed by the design team. Structured design, design walk-through and the incorporation of "antibugging" facilitate this testing by exposing and addressing problem areas before they become coded "bugs."

END
8-87
DTIC